Remarks:

Reconsideration of the application is respectfully requested in view of the foregoing amendments and following remarks. Claims 1-53 are pending in the application. No claims have been allowed. Claims 1, 6, 21, 27, 31, 34, 36, 37, 39, 41, 42, 46, 50, 52, and 53 are independent. Claims 1, 18, 27, 31, 34, 42-45, 48, and 53 have been amended, all for reasons not necessarily related to patentability. For example, claims 1 and 53 have been amended to address antecedent basis, and claim 31 has been amended to address claim form by replacing a comma with a period at the end of the claim. The amendments herein do not necessarily narrow the claims.

Interview Summary

Applicants thank the Examiner for her time during an interview on July 13, 2004. During the interview, claim 1 and Chilimbri were discussed. Agreement was reached that there is a distinction between Chilimbri and claim 1. To the extent that the Interview Summary uses language not identically appearing in claim 1 (e.g., "grouping and assigning memory locations based on DATA MEMBERS of the object class; NOT objects" rather than "responsive to said consulting metadata indicating how the data members of the object class are to be grouped into a plurality of separate groups comprising a first group and a second group, assigning memory locations for data members of the first group of the object class within a first unit of memory in the virtual memory system"), Applicants note that the claims are not limited by such language, but rather the actual language of each of the claims speaks for itself. Further, Applicants point out that discussion was limited to claim 1.

Information Disclosure Statement

Per the Examiner's request, a copy of the Information Disclosure Statement filed on March 14, 2001 has been submitted, accompanied by a copy of the postcard indicating receipt of the IDS by the Office on March 20, 2001.

Claim Objections

Claim 43 was objected to because of an informality. The term "method" has been amended to "one or more computer-readable media" to properly reference the amended claim 42 from which it depends. Therefore, Applicants respectfully request that the objection be removed.

Claim Rejections under § 112

The Action rejects claims 18-20, 27-30, and 47 under 35 U.S.C. § 112 as being indefinite. Claims 18, 27, and 47 have been amended. In claim 18, "the null set" has been amended to "a null set." In claim 27, "the threshold" has been amended to "the threshold framework." In claim 47, "the second entries the layout directive field is defined" has been amended to "the second object class comprises a layout directive field defined." Therefore, Applicants respectfully request that the rejections to claims 18, 27, and 47 and dependent claims 19-20 and 28-30 be withdrawn.

Claim Rejections under § 101

The Action rejects claims 42-45 under 35 U.S.C. § 101 as being directed to non-statutory subject matter. Claims 42-45 have been amended to state "one or more computer-readable media having computer-executable instructions implementing an execution engine." Thus, claim 42 and its dependent claims 43-45 are patentable.

Cited Art

"Using Generational Garbage Collection to Implement Cache-Conscious Data Placement" by Trishul Chilimbri and James Larus ("Chilimbri").

U.S. Patent No. 6,189,141 to Benitez et al. ("Benitez") is entitled "Control Path Evaluating Trace Designator with Dynamically Adjustable Thresholds for Activation of Tracing for High (HOT) Activity and Low (COLD) Activity of Flow Control."

U.S. Patent No. 6,631,496 to Li et al. ("Li") is entitled "System for Personalizing, Organizing and Managing Web Information."

U.S. Patent No. 6,324,620 to Christenson et al. ("Christenson") is entitled "Dynamic DASD Data Management and Partitioning Based on Access Frequency Utilization and Capacity."

Patentability of Claims 1-5, 14, 31-33, 37-38, and 52 over Chilimbri in view of Li under § 103

The Action rejects claims 1-5, 14, 31-33, 37-38, and 52 under 35 U.S.C. § 103(a) as being unpatentable over Chilimbri in view of Li. Applicants respectfully submit that the claims in their present form are allowable over the cited art. To establish a *prima facie* case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. (*See* MPEP § 2142.)

Motivations to combine or modify references must come from the references themselves or be within the body of knowledge in the art. (*See* MPEP § 2143.01.)

Claim 1

Claim 1 is directed to a method of arranging a plurality of data members of an object class in a virtual memory system having separately loadable units, and recites in part:

> responsive to said consulting metadata indicating how the *data members of the object class are to be grouped into a plurality of separate groups* comprising a first group and a second group, assigning memory locations for *data members of the first group of the object class within a first unit* of memory in the virtual memory system; and
>
> responsive to said consulting metadata indicating how the data members of the object class are to be grouped into a plurality of separate groups comprising a first group and a second group, assigning memory locations for data members of the *second group of the object class within a second unit of memory* in the virtual memory system separately loadable into primary memory from the first unit. (emphasis added)

For example, the application describes at page 20, line 3, et seq.:

> An overview of an exemplary data layout optimization method is shown in Figure 5. At 502, profile data for the object is collected. Examples of profile data are shown in the illustrated embodiments below. At 504, the *data members of an object* are grouped based on the profile data. Techniques for achieving such grouping are shown below. Then, at 506, at runtime, data members from the same groups are arranged at neighboring locations in the memory system; members from different groups are placed at locations separately loadable from each other (e.g., members from Group A are placed at locations in one page of memory, and members from Group B are placed at locations in another page of memory). (emphasis added)

Thus, different groups of data members for the same object are placed at separately loadable memory locations (see also, e.g., FIG. 8). For example, a data member of an object might refer to a property of the object, such as an integer field or an array (see application at page 10, lines 8-10, and page 13, lines 25-27). The Action rejects the claim based on Chilimbri. Applicants respectfully disagree.

Chilimbri's description of placing objects fails to teach or suggest grouping data members of an object class into a plurality of separate groups and assigning data members within first and second units of memory of a virtual memory system, the second unit being separately loadable from the first unit. In its rejection of claim 1, the Action relies on various passages in Chilimbri; however, these passages describe a scenario involving placing *objects*, not data members of the same object. Claim 1 recites "data members of the object class are to be grouped into a plurality of separate groups," "assigning memory locations for data members of the first group of the object class within a first unit," and "assigning memory locations for data members of the second group of the object class within a second unit of memory in the virtual memory system separately loadable into primary memory from the first unit."

For example, the Action relies upon Chilimbri at page 41, col. 1, par. 3, lines 1-6 to col. 2, par. 1, lines 1-5. During its discussion of layout of objects, Chilimbri states (as noted in the Action):

> As described in Section 3, generational garbage collection copies live objects to *TO space*. Our goal is to use data profiling information to produce a cache-conscious layout of objects in *TO space* that places *objects with high temporal affinity next to each other*, so that they are likely to be in the same cache block. The data profiling information captures the temporal ordering of base object addresses, which our system uses to construct object affinity graphs. An object affinity graph is a weighted undirected graph in which nodes represent objects and edges encode temporal affinity between objects. (emphasis added)

Thus, Chilimbri does describe "places objects . . . next to each other"; however, one of ordinary skill in the art could not be expected to surmise the claimed arrangement of "data members of the object class are to be grouped into a plurality of separate groups," "assigning memory locations for data members of the first group of the object class within a first unit," and "assigning memory locations for data members of the second group of the object class within a second unit of memory in the virtual memory system separately loadable into primary memory from the first unit" from the mere mention of placing objects next to each other.

Chilimbri also states at page 40, col. 2, par. 2:

> If most objects are small (< 32 bytes), then it is not necessary for data profiling to distinguish different fields within the same object, since cache blocks are currently larger (e.g., 64 bytes in the UltraSparc [25]) and growing. *Profiling can be implemented at object, not field, granularity.* Moreover, if most object accesses are not lightweight (i.e., multiple fields are accessed together or an access involves a method invocation), then profiling instrumentation (several instructions per object access) will not incur a large overhead. (emphasis added)

Thus, Chilimbri actually leads away from the claimed arrangement of "data members of the object class are to be grouped into a plurality of separate groups," "assigning memory locations for data members of the first group of the object class within a first unit," and "assigning memory locations for data members of the second group of the object class within a second unit of memory in the virtual memory system separately loadable into primary memory from the first unit" because it states that profiling can be implemented at object granularity.

Li similarly fails to teach or suggest "assigning memory locations for data members of the first group of the object class within a first unit" and "assigning memory locations for data members of the second group of the object class within a second unit of memory in the virtual memory system separately loadable into primary memory from the first unit," as recited in claim 1. As the Action notes, Li does describe the use of "user-specific metadata" (see col. 6, lines 25-28). However, one of ordinary skill in the art could not be expected to surmise the claimed arrangement of "assigning memory locations for data members of the first group of the object class within a first unit" and "assigning memory locations for data members of the second group of the object class within a second unit of memory in the virtual memory system separately loadable into primary memory from the first unit" from the mere mention of user-specific metadata.

Finally, Applicants can not find in Chilimbri or Li a suggestion to modify or combine the references to result in the claimed arrangement. Because the cited references, both alone and in combination, fail to teach or suggest the combination of elements recited in claim 1, Applicants believe the claim is not subject to a 103(a) rejection and request the rejection be withdrawn.

Thus, claim 1 and its dependent claims, 2-5, are allowable over the cited art.

Claim 14

Claim 14, discussed *infra*, is allowable over the cited art.

Claim 31

Claim 31 is directed to a computer-implemented method for optimizing the data layout of software comprising a set of object class definitions, and recites in part:

> *grouping the data members of the object class definitions into groups* according to how frequently the data members of the object class definitions are referenced in memory, wherein the groups comprise a more frequently referenced group and a less frequently referenced group (emphasis added)

Chilimbri and Li each fail to teach or suggest grouping the data members of the object class definitions into groups, much less according to how frequently the data members of the object class definitions are referenced in memory. Chilimbri does describe "a cache-conscious layout of objects," but one of ordinary skill in the art could not be expected to surmise the claimed arrangement of "grouping the data members of the object class definitions into groups" from the mere mention of layout of objects. Also, Li does describe the use of "user-specific metadata," but one of ordinary skill in the art could not be expected to surmise the claimed arrangement of "grouping the data members of the object class definitions into groups" from the mere mention of user-specific metadata.

Since the cited references, both alone and in combination, fail to teach or suggest the combination of elements recited in claim 31, Applicants believe the claim is not subject to a 103(a) rejection and request the rejection be withdrawn.

Thus, claim 31 and its dependent claims, 32 and 33, are allowable over the cited art.

Claim 37

Claim 37 is directed to a computer-implemented method for optimizing the layout of data members of an object class within a memory system comprising primary memory and secondary memory, and recites in part:

> *dividing the data members* into a hot group and a cold group based on a threshold edge cost (emphasis added)

Chilimbri and Li each fail to teach or suggest dividing the data members, much less dividing the data members into a hot group and a cold group based on a threshold edge cost. Chilimbri does describe "a cache-conscious layout of objects," but one of ordinary skill in the art could not be expected to surmise the claimed arrangement of "dividing the data members into a hot group and a cold group" from the mere mention of layout of objects. Also, Li does describe the use of "user-specific metadata," but one of ordinary skill in the art could not be expected to surmise the claimed arrangement of "dividing the data members into a hot group and a cold group" from the mere mention of user-specific metadata.

Since the cited references, both alone and in combination, fail to teach or suggest the combination of elements recited in claim 37, Applicants believe the claim is not subject to a 103(a) rejection and request the rejection be withdrawn.

Thus, claim 37 and its dependent claim, 38, are allowable over the cited art.

Claim 52

Claim 52 is directed to a method of laying out data members of an object having a plurality of data members, and recites in part:

> responsive to consulting the metadata, *arranging the data members of the object into a plurality of separately loadable blocks* to group at least one more frequently accessed member into a first one of the plurality of separately loadable blocks and at least one less frequently accessed data member into a second one of the plurality of separately loadable blocks (emphasis added)

Chilimbri and Li each fail to teach or suggest arranging the data members of the object into a plurality of separately loadable blocks, much less arranging the data members to group at least one more frequently accessed member into a first one of the plurality of separately loadable blocks and at least one less frequently accessed data member into a second one of the plurality of separately loadable blocks. Chilimbri does describe "a cache-conscious layout of objects,"

but one of ordinary skill in the art could not be expected to surmise the claimed arrangement of "arranging the data members of the object into a plurality of separately loadable blocks" from the mere mention of layout of objects. Also, Li does describe the use of "user-specific metadata," but one of ordinary skill in the art could not be expected to surmise the claimed arrangement of "arranging the data members of the object into a plurality of separately loadable blocks" from the mere mention of user-specific metadata.

Since the cited references, both alone and in combination, fail to teach or suggest the combination of elements recited in claim 52, Applicants believe the claim is not subject to a 103(a) rejection and request the rejection be withdrawn.

Patentability of Claims 6-13, 15-21, 36, 39-51, and 53 over Chilimbri under § 102(b)

The Action rejects claims 6-13, 15-21, 36, 39-51, and 53 under 35 U.S.C. § 102(b) as being anticipated by Chilimbri. Applicants respectfully submit that the claims in their present form are allowable over the cited art. For a 102(b) rejection to be proper, the cited art must show each and every element as set forth in a claim. (*See* MPEP § 2131.01.) However, the cited art does not so show. For example, with respect to claim 6, Chilimibri does not teach or suggest grouping the more frequently referenced data members out of the plurality of data members of the object together.

Claim 6

Claim 6 is directed to a method of arranging a plurality of data members of an object in a memory system, and recites in part:

> responsive to said identifying the more frequently referenced data members out of the plurality of data members of the object, *grouping the more frequently referenced data members* out of the plurality of data members of the object together into a group of more frequently referenced data members (emphasis added)

Chilimbri fails to teach or suggest grouping the more frequently referenced data members out of the plurality of data members of the object together into a group of more frequently referenced data members. In its rejection of claim 6, the Action relies on various passages in Chilimbri; however, these passages describe a scenario involving layout of objects,

not "grouping the more frequently referenced data members out of the plurality of data members of the object together" as recited in claim 6.

For example, the Action relies upon Chilimbri at page 41, col. 1, par. 3, lines 1-6 to col. 2, par. 1, lines 1-5. During its discussion of layout of objects, Chilimbri states (as noted in the Action):

> As described in Section 3, generational garbage collection copies live objects to *TO space. Our goal is to use data profiling information to produce a cache-conscious layout of objects in TO space that places objects with high temporal affinity next to each other*, so that they are likely to be in the same cache block. The data profiling information captures the temporal ordering of base object addresses, which our system uses to construct object affinity graphs. An object affinity graph is a weighted undirected graph in which nodes represent objects and edges encode temporal affinity between objects. (emphasis added)

Thus, Chilimbri does describe "a cache-conscious layout of objects"; however, one of ordinary skill in the art could not be expected to surmise the claimed arrangement of "grouping the more frequently referenced data members out of the plurality of data members of the object together" from the mere mention of layout of objects.

Applicants fail to see how the above passage would lead one of ordinary skill in the art to the claimed arrangement, which involves "grouping the more frequently referenced data members out of the plurality of data members of the object together" as recited in claim 6.

Since the cited reference fails to describe at least one element recited in claim 6, Applicants believe the claim is not subject to a 102(b) rejection and request the rejection be withdrawn.

Thus, claim 6 and its dependent claims, 7-20, are allowable over the cited art.

Claim 21

Claim 21 is directed to a method of arranging a plurality of data members of an object in a memory system by separating the plurality of data members, and recites in part:

> responsive to said identifying the more frequently referenced data members out of the plurality of data members of the object and less frequently referenced data members out of the plurality of data members of the object, *grouping the more frequently referenced data members* out of the plurality of data members of the object together into first group of data members *and the less frequently referenced data members* out of the plurality of data members of the object together into a second group of data members (emphasis added)

Chilimbri fails to teach or suggest grouping the more frequently referenced data members and the less frequently referenced data members out of the plurality of data members of the object. Chilimbri does describe "a cache-conscious layout of objects," but one of ordinary skill in the art could not be expected to surmise the claimed arrangement of "grouping the more frequently referenced data members out of the plurality of data members of the object" from the mere mention of layout of objects.

Since the cited reference fails to describe at least one element recited in claim 21, Applicants believe the claim is not subject to a 102(b) rejection and request the rejection be withdrawn.

Thus, claim 21 and its dependent claims, 22-26, are allowable over the cited art.

Claim 36

Claim 36 is directed to a computer-implemented method for optimizing the layout of data members of an object class within a memory system comprising primary memory and secondary memory, and recites in part:

> *separating the data members into a hot group and a cold group* based on how frequently memory references are made to the data members defined for the object during profiling (emphasis added)

Chilimbri fails to teach or suggest separating the data members into a hot group and a cold group. Chilimbri does describe "a cache-conscious layout of objects," but one of ordinary skill in the art could not be expected to surmise the claimed arrangement of "separating the data members into a hot group and a cold group" from the mere mention of layout of objects.

Since the cited reference fails to describe at least one element recited in claim 36, Applicants believe the claim is not subject to a 102(b) rejection and request the rejection be withdrawn.

Thus, claim 36 is allowable over the cited art.

Claim 39

Claim 39 is directed to a method of monitoring affinity among a plurality of data members of an object class in a memory system, and recites in part:

> computing an affinity matrix for the object class with a dot product operation, wherein the affinity matrix indicates an affinity between a first data member of the object and a second data member of the object

Chilimbri fails to teach or suggest an affinity matrix that indicates an affinity between a first data member of the object and a second data member of the object. In its rejection of claim 6, the Action relies on various passages in Chilimbri; however, these passages describe a scenario involving object affinity graphs, not an affinity matrix that "indicates an affinity between a first data member of the object and a second data member of the object" as recited in claim 39.

For example, the Action relies upon Chilimbri at page 41, col. 1, par. 2, lines 1-9. During its discussion of object affinity graphs, Chilimbri states (as noted in the Action):

> The object access buffer is normally processed just before a scavenge to *construct object affinity graphs* (Section 5). However, it may overflow between scavenges. Rather than include an explicit overflow check in the instrumentation, the virtual memory system causes a page trap on buffer overflow. The trap handler processes the buffer to *construct object affinity graphs* and restarts the application. Our experience indicates that setting the buffer size to 15,000 entries (60 KB) prevents overflow. (emphasis added)

Thus, Chilimbri does describe "object affinity graphs"; however, one of ordinary skill in the art could not be expected to surmise the claimed arrangement of an affinity matrix that "indicates an affinity between a first data member of the object and a second data member of the object" from the mere mention of object affinity graphs.

Applicants fail to see how the above passage would lead one of ordinary skill in the art to the claimed arrangement, which involves an affinity matrix that "indicates an affinity between a first data member of the object and a second data member of the object" as recited in claim 39.

Since the cited reference fails to describe at least one element recited in claim 39, Applicants believe the claim is not subject to a 102(b) rejection and request the rejection be withdrawn.

Thus, claim 39 and its dependent claim, 40, are allowable over the cited art.

Claim 41

Claim 41 is directed to a system for performing data layout optimization on software comprising a set of object class definitions, and recites in part:

> a *data member grouper* operative to consult the profile data and group the data members of an object into at least two groups, wherein the groups comprise a more frequently referenced group of data members and a less frequently referenced group of data members, wherein the data member grouper is further operative to annotate the object class definitions to indicate into which of the groups data members defined in the object class definitions fall (emphasis added)

Chilimbri fails to teach or suggest a data member grouper operative to consult the profile data and group the data members of an object into at least two groups. As discussed *supra*, Chilimbri does describe "a cache-conscious layout of objects," but one of ordinary skill in the art could not be expected to surmise the claimed arrangement of "a data member grouper operative to consult the profile data and group the data members of an object into at least two groups" from the mere mention of layout of objects.

Since the cited reference fails to describe at least one element recited in claim 41, Applicants believe the claim is not subject to a 102(b) rejection and request the rejection be withdrawn.

Thus, claim 41 is allowable over the cited art.

Claim 42

Claim 42 is directed to an execution engine operable to perform operations on an object of an object class defined by an object class definition and residing in a virtual memory system having units separately loadable into primary memory, and recites in part:

> means for *arranging the plurality of data members* of the object of the object class within plural units of memory in the virtual memory system according to the information about the plurality of data members of the object class, wherein the means for arranging is operative to consult the information indicating information about the data members responsive to the request to arrange (emphasis added)

Chilimbri fails to teach or suggest means for arranging the plurality of data members of the object of the object class within plural units of memory in the virtual memory system. As discussed *supra*, Chilimbri does describe "a cache-conscious layout of objects," but one of ordinary skill in the art could not be expected to surmise the claimed arrangement of "means for arranging the plurality of data members of the object of the object class within plural units of memory in the virtual memory system" from the mere mention of layout of objects.

Since the cited reference fails to describe at least one element recited in claim 42, Applicants believe the claim is not subject to a 102(b) rejection and request the rejection be withdrawn.

Thus, claim 42 and its dependent claims, 43-45, are allowable over the cited art.

Claim 46

Claim 46 is directed to a data structure for indicating how a plurality of data members defined by an object class definition of an object class are to be arranged in a memory system at runtime of an object, and recites in part:

> for at least two data members of the object class, a grouping field indicating a group into which data members are to be placed at runtime, wherein the grouping field for a first at least one data member of the object class indicates the at first at least one data member of the object class is to be placed into a first group residing in a first unit of the memory system at runtime, and the grouping field for a second at least one data member of the object class indicates the at second at least one data member of the object class is to be placed into a second group residing in a second unit of the memory system separately-loadable from the first unit at runtime

Chilimbri fails to teach or suggest a grouping field indicating a group into which data members are to be placed at runtime. As discussed *supra*, Chilimbri does describe "a cache-conscious layout of objects," but one of ordinary skill in the art could not be expected to surmise the claimed arrangement of "a grouping field indicating a group into which data members are to be placed at runtime" from the mere mention of layout of objects.

Since the cited reference fails to describe at least one element recited in claim 46, Applicants believe the claim is not subject to a 102(b) rejection and request the rejection be withdrawn.

Thus, claim 46 and its dependent claims, 47-49, are allowable over the cited art.

Claim 50

Claim 50 is directed to a set of data structures for a respective set of object classes, and recites in part:

> grouping fields for the data members of the respective object class, the grouping fields indicating how data members of the respective object are to be divided into the separately-loadable units of the memory system at runtime.

Chilimbri fails to teach or suggest grouping fields for the data members of the respective object class. As discussed *supra*, Chilimbri does describe "a cache-conscious layout of objects," but one of ordinary skill in the art could not be expected to surmise the claimed arrangement of "grouping fields for the data members of the respective object class" from the mere mention of layout of objects.

Since the cited reference fails to describe at least one element recited in claim 50, Applicants believe the claim is not subject to a 102(b) rejection and request the rejection be withdrawn.

Thus, claim 50 and its dependent claim, 51, are allowable over the cited art.

Claim 53

Claim 53 is directed to a method of dividing an object comprising a plurality of data members into plural units of memory in the memory system, and recites in part:

> responsive to said observing, loading the more referenced data members into a first unit of memory in the memory system;

Chilimbri fails to teach or suggest loading the more referenced data members into a first unit of memory in the memory system. As discussed *supra*, Chilimbri does describe "a cache-conscious layout of objects," but one of ordinary skill in the art could not be expected to surmise the claimed arrangement of "loading the more referenced data members into a first unit of memory in the memory system" from the mere mention of layout of objects.

Since the cited reference fails to describe at least one element recited in claim 53, Applicants believe the claim is not subject to a 102(b) rejection and request the rejection be withdrawn.

Thus, claim 53 is allowable over the cited art.

Patentability of Claims 22 and 25 over Chilimbri in view of Benitez under § 103

The Action rejects claims 22 and 25 under 35 U.S.C. § 103(a) as being unpatentable over Chilimbri in view of Benitez. Applicants respectfully submit that claims 22 and 25 are directly or indirectly dependent on independent claim 21, the features of which were discussed above with respect to Chilimbri. Although Applicants could argue various features of the claims, in the interest of brevity, Applicants point out that Benitez does not supplement the shortcomings of Chilimbri with respect to claim 21. Therefore, claim 21 is allowable over Chilimbri in light of Benitez, and so are dependent claims 22 and 25.

Patentability of Claims 23, 24, and 26 over Chilimbri in view of Benitez and further in view of Christenson under § 103

The Action rejects claims 23, 24, and 26 under 35 U.S.C. § 103(a) as being unpatentable over Chilimbri in view of Benitez and further in view of Christenson. Applicants respectfully submit that claims 23, 24, and 26 are directly or indirectly dependent on independent claim 21, the features of which were discussed above with respect to Chilimbri. Although Applicants could argue various features of the claims, in the interest of brevity, Applicants point out that Christenson does not supplement the shortcomings of Chilimbri with respect to claim 21. Therefore, claim 21 is allowable over Chilimbri in light of Christenson, and so are dependent claims 23, 24, and 26.

Patentability of Claims 27, 29-30, and 34-35 over Benitez under § 102(e)

The Action rejects claims 27, 29-30, and 34-35 under 35 U.S.C. § 102(e) as being anticipated by Benitez. Applicants respectfully submit that the claims in their present form are allowable over the cited art. For a 102(e) rejection to be proper, the cited art must show each and every element as set forth in a claim. (*See* MPEP § 2131.01.) However, the cited art does not so show. For example, with respect to claim 27, Benitez does not teach or suggest assembling a list of the data members indicating how many times each data member in the list was referenced in memory.

Claim 27

Claim 27 is directed to a method of splitting data members of an object class into a set of groups to be placed at separately-loadable units of memory in a memory system by consulting profile data indicating how many times the data members were referenced in memory, and recites in part:

> assembling a list of the data members indicating how many times each data member in the list was referenced in memory

Benitez fails to teach or suggest assembling a list of the data members indicating how many times each data member in the list was referenced in memory. In its rejection of claim 27, the Action relies on various passages in Benitez; however, these passages describe a scenario

involving identification and evaluation of traces, not "assembling a list of the data members indicating how many times each data member in the list was referenced in memory" as recited in claim 27.

For example, the Action relies upon Benitez at col.2, lines 35-52. During its discussion of identifying and evaluating traces, Benitez states (as noted in the Action):

> The present invention is a system, method, and product for continuous path evaluation at run time in order to *identify and evaluate hot traces*. In one aspect of the invention, a control-path-evaluating trace designator is disclosed. The control-path-evaluating trace designator designates at least one hot trace, if present, in the original instructions of the executable file. A hot trace is a trace through which control frequently passes, or has passed more than a predetermined number of times, as determined in accordance with the invention and described below. For convenience, the term "frequent," and its grammatical variants, are used herein to refer both to control passing through instructions either at a rate, or for a number of times (sometimes referred to as the magnitude of occurrences), greater than a threshold value. In one embodiment, such threshold is predetermined. In alternative embodiments, such threshold is dynamically selected or adjusted (hereafter, simply "dynamically adjusted"). (emphasis added)

Benitez also states at col. 2, lines 53-59:

> A trace typically is made up of one or more blocks of original instructions of an executable file, each of which may be reached through a common control path. A block is made up of one or more basic blocks. A basic block typically is a sequence of instructions of an executable file such that there is only one entrance into the basic block and such entrance is the first instruction in the sequence.

Thus, Benitez does describe identifying and evaluating executable instructions; however, one of ordinary skill in the art could not be expected to surmise the claimed arrangement of "assembling a list of the data members indicating how many times each data member in the list was referenced in memory" from the mere mention of identifying and evaluating executable instructions.

Since the cited reference fails to describe at least one element recited in claim 27, Applicants believe the claim is not subject to a 102(e) rejection and request the rejection be withdrawn.

Thus, claim 27 and its dependent claims, 28-30, are allowable over the cited art.

Claim 34

Claim 34 is directed to a method of specifying data layout optimization for software comprising a set of objects belonging to object classes, and recites in part:

> profiling the software to generate profile data indicating how many times each data member of each of the set of objects was referenced in memory during the profiling

Benitez fails to teach or suggest a data member grouper operative to consult the profile data and group the data members of an object into at least two groups. As discussed *supra*, Benitez does describe identifying and evaluating executable instructions, but one of ordinary skill in the art could not be expected to surmise the claimed arrangement of "assembling a list of the data members indicating how many times each data member in the list was referenced in memory" from the mere mention of identifying and evaluating executable instructions.

Since the cited reference fails to describe at least one element recited in claim 34, Applicants believe the claim is not subject to a 102(e) rejection and request the rejection be withdrawn.

Thus, claim 34 and its dependent claim, 35, are allowable over the cited art.

Patentability of Claim 28 over Benitez in view of Christenson under § 103

The Action rejects claim 28 under 35 U.S.C. § 103(a) as being unpatentable over Chilimbri in view of Christenson. Applicants respectfully submit that claim 28, which depends from claim 27, the features of which were discussed above with respect to Benitez. Although Applicants could argue various features of the claims, in the interest of brevity, Applicants point out that Christenson does not supplement the shortcomings of Benitez with respect to claim 27. Therefore, claim 27 is allowable over Benitez in light of Christenson, and so is dependent claim 28.

Request for Interview

If any issues remain, the Examiner is formally requested to contact the undersigned attorney prior to issuance of the next Office Action in order to arrange a telephonic interview. It is believed that a brief discussion of the merits of the present application may expedite prosecution. Applicants submit the foregoing formal Amendment so that the Examiner may fully evaluate Applicants' position, thereby enabling the interview to be more focused.

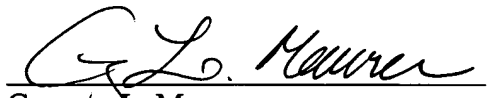
This request is being submitted under MPEP § 713.01, which indicates that an interview may be arranged in advance by a written request.

Conclusion

The claims in their present form should now be allowable. Such action is respectfully requested.

Respectfully submitted,

KLARQUIST SPARKMAN, LLP

By 
Gregory L. Maurer
Registration No. 43,781

One World Trade Center, Suite 1600
121 S.W. Salmon Street
Portland, Oregon 97204
Telephone: (503) 226-7391
Facsimile: (503) 228-9446